


FORM PTO-1390 U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE (REV-9-2001)		ATTORNEY'S DOCKET NUMBER DVME-1024US
TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371		U.S. APPLICATION NO. (if any) 10/089918
International Application No. PCT/NL00/00720	International Filing Date 06 October 2000 ✓	Priority Date Claimed 08 October 1999 ✓
TITLE OF INVENTION: Method for Transferring A Software Module From A Sender To A Receiver In A Computer System or Network		
APPLICANT(S) FOR DO/EO/US: BOLWIDT, Erwin, Joost		
Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:		
<ol style="list-style-type: none"> 1. <input checked="" type="checkbox"/> This is a First submission of items concerning a filing under 35 U.S.C. 371. 2. <input type="checkbox"/> This is a Second or Subsequent submission of items concerning a filing under 35 U.S.C. 371. 3. <input type="checkbox"/> This is an express request to begin national examination procedures (35 U.S.C. 371(f)). The submission must include items (5), (6), (9) and (21) indicated below. 4. <input checked="" type="checkbox"/> The US has been elected by the expiration of 19 months from the priority date (Article 31). 5. <input checked="" type="checkbox"/> A copy of the International Application as filed (35 U.S.C. 371(c)(2)) <ol style="list-style-type: none"> a. <input type="checkbox"/> is attached hereto (required only if not communicated by the International Bureau). b. <input checked="" type="checkbox"/> has been communicated by the International Bureau. c. <input type="checkbox"/> is not required, as the application was filed in the United States Receiving Office (RO/US). 6. <input type="checkbox"/> An English language translation of the International Application as filed (35 U.S.C. 371(c)(2)). <ol style="list-style-type: none"> a. <input type="checkbox"/> is attached hereto. b. <input type="checkbox"/> has been previously submitted under 35 U.S.C. 154(d)(4). 7. <input type="checkbox"/> Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371(c)(3)) <ol style="list-style-type: none"> <input type="checkbox"/> are attached hereto (required only if not communicated by the International Bureau) <input type="checkbox"/> have been communicated by the International Bureau. <input type="checkbox"/> have not been made; however, the time limit for making such amendments has NOT expired. <input type="checkbox"/> have not been made and will not be made. 8. <input type="checkbox"/> An English language translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)). 9. <input checked="" type="checkbox"/> An oath or declaration of the inventor(s) (35 U.S.C. 371(c)(4)) (unsigned). 10. <input type="checkbox"/> An English language translation of the annexes of the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371(c)(5)) 		
Items 11 to 20 below concern document(s) or information included:		
<ol style="list-style-type: none"> 11. <input checked="" type="checkbox"/> An Information Disclosure Statement under 37 CFR 1.97 and 1.98 and cited references. 12. <input type="checkbox"/> An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included. 13. <input checked="" type="checkbox"/> A FIRST preliminary amendment. 14. <input type="checkbox"/> A SECOND or SUBSEQUENT preliminary amendment. 15. <input type="checkbox"/> A substitute specification. 16. <input type="checkbox"/> A change or power of attorney and/or address letter. 17. <input type="checkbox"/> A computer-readable form of the sequence listing in accordance with PCT Rule 13 ter.2 and 35 U.S.C. 1.821-1.825. 18. <input type="checkbox"/> A second copy of the published international application under 35 U.S.C. 154(d)(4). 19. <input type="checkbox"/> A second copy of the English language translation of the international application under 35 U.S.C. 154(d)(4). 20. <input checked="" type="checkbox"/> Other items or information: Certificate of Express Mail/Copy of International Preliminary Examination Report with annexes (amended pages of specification and claims) 		

KEVIN J. DUNLEAVY
KNOBLE & YOSHIDA, LLC
Suite 1350
1628 John F. Kennedy Blvd.
Philadelphia, Pennsylvania 19103
Telephone: (215) 599-0600
Facsimile: (215) 599-0601
E-mail: kjdunleavy@patentwise.com


Signature

Kevin J. Dunleavy

Kevin J. Dunleavy
Name
32,024
Registration Number

PATENT
DVME-1024US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
ACTING AS THE DESIGNATED/ELECTED OFFICE

In re Application of:)	
Erwin Joost Bolwidt)	Group Art Unit: Not Yet Assigned
)	
Serial No.: Not Yet Assigned)	Examiner: Not Yet Assigned
)	
Filed: Concurrently Herewith)	International Application No.:
)	PCT/NL00/00720
Title: Method for Transferring A Software)	
Module From A Sender To A Receiver In A)	International Filing Date:
Computer System or Network)	06 October 2000

FIRST PRELIMINARY AMENDMENT

Commissioner For Patents
BOX PCT
Washington, D.C. 20231

Sir:

Prior to the calculation of fees for the above-captioned application, please amend the application as follows:

Certificate of Express Mail

Date: April 4, 2002

Express Mail Label No.: EL606862685US

I hereby certify that this paper, along with any document or paper referred to as being attached, is being deposited with the United States Postal Service "EXPRESS MAIL POST OFFICE TO ADDRESSEE" service under 37 CFR 1.10 in an envelope addressed to the Commissioner for Patents, BOX PCT, Washington, D. C. 20231.

Lynne Webb
Name of person mailing correspondence

Lynne Webb
Signature of person mailing correspondence

In the Claims:

Please amend claims 3-6 and 8 to read as follows:

3. (Amended) Method according to claim 1, wherein the receiver obtains the software module to be transferred by combining the object received with the class or group of classes retrieved from its database or received, wherein the receiver transmits a message "transfer succeeded" or "transfer not succeeded" depending on whether or not the receiver succeeds in combining the object and class or group of classes.
4. (Amended) Method according to claim 1, wherein the receiver stores each class and group of classes with the corresponding class identifier received in its database for later use.
5. (Amended) Method according to claim 1, wherein a sender further combines the length of the data file of the class or group of classes with the given name and the result of the hash function to provide the class identifier.
6. (Amended) Method according to claim 1, wherein the receiver checks a class or group of classes received from a sender by comparing the result of the hash function of the received class identifier with the result obtained by carrying out the same cryptographic hash function on the data file of the class or group of classes received.
8. (Amended) Method according to claim 1, wherein senders and receivers are computers in a computer network.

Please add new claims 10-26 as follows:

- 10. (New) Method according to claim 2, wherein the receiver obtains the software module to be transferred by combining the object received with the class or group of classes retrieved from its database or received, wherein the receiver transmits a message "transfer

succeeded” or “transfer not succeeded” depending on whether or not the receiver succeeds in combining the object and class or group of classes.

11. (New) Method according to claim 2, wherein the receiver stores each class and group of classes with the corresponding class identifier received in its database for later use.

12. (New) Method according to claim 3, wherein the receiver stores each class and group of classes with the corresponding class identifier received in its database for later use.

13. (New) Method according to claim 2, wherein a sender further combines the length of the data file of the class or group of classes with the given name and the result of the hash function to provide the class identifier.

14. (New) Method according to claim 3, wherein a sender further combines the length of the data file of the class or group of classes with the given name and the result of the hash function to provide the class identifier.

15. (New) Method according to claim 4, wherein a sender further combines the length of the data file of the class or group of classes with the given name and the result of the hash function to provide the class identifier.

16. (New) Method according to claim 2, wherein the receiver checks a class or group of classes received from a sender by comparing the result of the hash function of the received class identifier with the result obtained by carrying out the same cryptographic hash function on the data file of the class or group of classes received.

17. (New) Method according to claim 3, wherein the receiver checks a class or group of classes received from a sender by comparing the result of the hash function of the received class identifier with the result obtained by carrying out the same cryptographic hash function on the data file of the class or group of classes received.

19. (New) Method according to claim 5, wherein the receiver checks a class or group of classes received from a sender by comparing the result of the hash function of the received class identifier with the result obtained by carrying out the same cryptographic hash function on the data file of the class or group of classes received.

20. (New) Method according to claim 2, wherein senders and receivers are computers in a computer network.

21. (New) Method according to claim 3, wherein senders and receivers are computers in a computer network.

22. (New) Method according to claim 4, wherein senders and receivers are computers in a computer network.

23. (New) Method according to claim 5, wherein senders and receivers are computers in a computer network.

24. (New) Method according to claim 6, wherein senders and receivers are computers in a computer network.

25. (New) Method according to claim 7, wherein senders and receivers are computers in a computer network.

26. (New) Method according to claim 8, wherein the computer network is the Internet. --

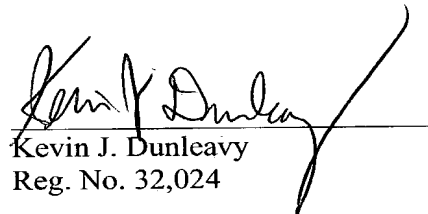
Remarks

Claims 1-9 filed on October 25, 2001, and annexed to the International Preliminary Examination Report for the above-identified application are currently pending and form the basis for this First Preliminary Amendment.

Claims 2-6 and 8 have been amended. Claims 10-26 have been added. Claims 1-26 are pending for examination as a result of entry of this First Preliminary Amendment.

This preliminary amendment has eliminated multiple dependent claims without prejudice to resubmission and has corrected some minor typographical errors in the claims. Favorable consideration and entry of the amendment is requested.

Respectfully submitted,


Kevin J. Dunleavy
Reg. No. 32,024

Dated: April 4, 2002

KNOBLE & YOSHIDA LLC
Eight Penn Center, Suite 1350
1628 John F. Kennedy Blvd.
Philadelphia, PA 19103
Telephone: (215) 599-0600
Facsimile: (215) 599-0601
e-mail: kjdunleavy@patentwise.com

Method for transferring a software module from a sender to a receiver in a computer system or network

The invention relates to a method for transferring a software module from a sender to a receiver in a computer system or network, wherein the software module comprises at least one object and at least one class, the object being an instance of the class(es).

In object-oriented software technology it is known to build a software module as a combination of so-called objects and classes, wherein the or each object that belongs to a class, is called an instance of the class. The objects generally contain only particular values for the variables specific to a predetermined software module, wherein the variables and methods to be carried out by the software module are defined in the class or classes.

With increasing use of computer systems and networks, such as the Internet, there is an increasing transfer of software modules of the object-oriented type between processes executed within one computer system or between computers of a computer network. This increasing transferring of software modules results in an increase of data traffic within the computer system of computer network.

A first object of the present invention is to provide a method of the above-mentioned type, wherein the data traffic within a computer system or computer network during transferring a software module is reduced.

It is a further object of the invention to provide a method of this type, wherein a software module can be transferred in a secure manner.

According to the invention a method of the above-mentioned type is provided, wherein each class or group of classes is provided with a class identifier, wherein both the sender and receiver comprises a database of classes and groups of classes with corresponding class identifiers, wherein the

Rec'd PCT/PTO 12 NOV 2002

Docket No. DVME-1024US

2

10/089918

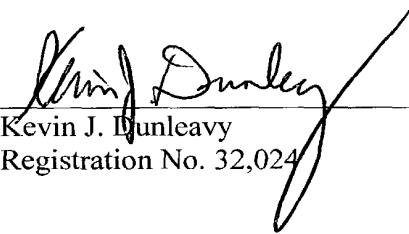
1. Declaration and Power of Attorney for Erwin Joost Bolwidt;
2. Copy of the Notification of Missing Requirements....;
3. Check in the amount of \$65.00.

A check in the amount of \$65.00 is enclosed for the surcharge for the late filing of the Declaration and Power of Attorney. Applicant is entitled to claim small entity status.

The Commissioner is authorized to charge any additional fees associated with this response or credit any overpayment, to Deposit Account No. 50-0462.

Respectfully submitted,

Date: 11/12/02


Kevin J. Dunleavy
Registration No. 32,024

Customer No. 21302
KNOBLE & YOSHIDA, LLC
Eight Penn Center, Suite 1350
1628 John F. Kennedy Blvd.
Philadelphia, PA 19103
Tel: (215) 599-0600
Fax: (215) 599-0601
E-mail: KJDunleavy@patentwise.com

11/18/2002 HKAYPAGH 00000046 10089918

01 FC:2617

65.00 OP

Rec'd PCT/PTO Q 4 APR 2002

10/089918

WO3473-dv/rp

Method for transferring a software module from a sender to a receiver in a computer system or network

The invention relates to a method for transferring a software module from a sender to a receiver in a computer system or network, wherein the software module comprises at least one object and at least one class, the object being an instance of the class(es), wherein each class or group of classes is provided with a class identifier, wherein both the sender and receiver comprises a database of classes and groups of classes with corresponding class identifier, wherein the sender transmits the class identifier of a software module to be transferred to the receiver and the receiver checks its database for presence of the received class identifier, and wherein the sender transfers only the object of the software module or both the object and class or group of classes depending on the presence or absence of the class or group of classes at the receiver method for transferring a software module from a sender to a receiver in a computer system or network, wherein the software module comprises at least one object and at least one class, the object being an instance of the class(es).

In object-oriented software technology it is known to build a software module as a combination of so-called objects and classes, wherein the or each object that belongs to a class, is called an instance of the class. The objects generally contain only particular values for the variables specific to a predetermined software module, wherein the variables and methods to be carried out by the software module are defined in the class or classes.

With increasing use of computer systems and networks, such as the Internet, there is an increasing transfer of software modules of the object-oriented type between processes executed within one computer system or between computers of a computer network. This increasing transferring of software modules results in an increase of data traffic within the computer system or computer network.

IBM, CRYSTALITZ, GENERAL MAGIC, GMD FOCUS, 'Mobile Agent Facility Specification', OMG TC Document, 2 June 1997, discloses a common conceptual model for differing mobile agent systems. To implement the transfer of classes, the class must be transferred from the source agent system if it does not exist at the destination agent system. One possible approach is the transfer of a list of the names of all possible classes with the agent creation or transfer request. The destination agent system requests only the classes on that list that it has not cached. The agent is transferred in serialised form, which is able to identify and verify the classes. Agent authenticators are used to provide a secure communications infrastructure. However, an attacker can monitor communications traffic that transports agents and decodes their state data. To counter this attack an agent may demand confidentiality services as a condition for transport. This increases the data traffic within the network.

A first object of the present invention is to provide a method of the above-mentioned type, wherein the data traffic within a computer system or computer network during transferring a software module is reduced.

It is a further object of the invention to provide a method of this type, wherein a software module can be transferred in a secure manner.

According to the invention a method of the above-mentioned type is provided, characterised in that the receiver transmits a message "present" or "absent" to the sender, and the sender provides each class or group of classes by combining a given name of each class or group of classes of a software module and the result of a cryptographic hash function, wherein said result is obtained by executing a cryptographic hash function on the data file of the class or group of classes.

In this manner, a method is obtained, wherein data traffic during transferring software modules is significantly reduced as the classes or groups of classes need not to be transferred in all transfers of software modules, and a secure identifier is obtained, wherein errors due to identical identifiers for different classes or groups of classes are excluded.

Figs. 2 and 3 show flow diagrams of the operation of a

sender and a receiver in the method of the invention.

A preferred embodiment of the method of the invention will be described as implemented by way of example in so-called software agents which can be used in a computer network, such as the Internet, for example for searching information on the Internet. However, it will be understood that the method of the invention is not restricted to this specific application. The method can be applied for transferring any software module of an object-oriented type as described.

Fig. 1 shows in a very schematic way the Internet comprising a number of interconnected computers or servers 1 and computers 2 of users. A user computer 2 can be connected to the Internet through a server 1 of an Internet service provider. If a user wishes to obtain information on a specific subject from the Internet, he can send a software agent with his request on the Internet to obtain this information. Finding adequate information on the Internet is a problem in view of the huge amount of information available on the Internet. Existing technology such as search engines store the most relevant information from the complete Internet to find this information in a quick manner if a user requests such information. Existing search engines show the disadvantage that searching the complete Internet from one location is time consuming so that such a search will not be carried out frequently and thereby the information stored by the search engine is dated fast. Moreover, existing search engines use their own criteria which do not necessarily correspond with the criteria which a user would use. A software agent searching the Internet with a request of its user may however use its own criteria for finding relevant information and the information can be obtained by communicating with other software agents also searching for information. During searching the Internet the software agents are transferred from one computer 1 to another computer 1 which causes a load of the network due to the data traffic involved in such transfers.

In order to reduce the load of the network caused by transferring the software agent, the following method is used.

It is noted that in the present description the term

sender is used to indicate a computer 1 or 2 from which a software agent is to be transferred to another computer 1 or 2 which receiving computer is called receiver. It is noted however that the present method can also be used to transfer an agent being present in a directly executable format in a process executed in one computer to another process executed in the same computer. In this case the terms sender and receiver refer to such processes between which a software agent is transferred. It is further noted that in an application of the method in a computer network, such as the Internet, it is not necessary that all computers of the network are adapted or programmed to operate as sender and/or receiver.

Each software agent is made as a software module of an object-oriented type. This means within the present description that the software module comprises one or more objects and one or more classes, wherein the objects are instances of the class or classes. An object comprises data, such as variable values. A class comprises definitions of the data structure, i.e. name and type of data fields, and definitions of functions or routines which are carried out on the objects of the class. Objects or more particular the variable values of the objects often change due to the execution of the routines which are defined in the class. If a software module comprises two or more classes these classes will be indicated as a group of classes in this specification.

According to the method described, a software module is transferred from a first computer 1, the sender, to a second computer 1, the receiver, in the computer network, by transferring the class or classes of the software module only if the class or classes are not present at the receiver side. In those computers of the network adapted to function as a sender and/or receiver a database is provided for storing classes or groups of classes, wherein a unique class identifier is added to each class or group of classes. When a sender wishes to transfer a software module to a receiver, the sender and receiver start to communicate to check or whether or not the classes or group of classes of the software module to be transferred is present in the receiver. The steps of the method are schematically shown

in figs. 2 and 3 for the sender and receiver sides, respectively.

To start the transfer protocol, the sender transmits a start message to the receiver indicating "hereinafter follow all data of a software module". All objects which are part of the software module to be transferred are combined in a data file and this data file is transmitted to the receiver. Further, the sender transmits the unique class identifier to the receiver. Thereafter, the receiver checks its database for the presence of the unique class identifier received. The receiver transmits a return message "present" or "absent" to the sender and the sender transmits a data file with the class or group of classes only if a message "absent" was received. If Java is used as software language for the software module, the data files of the classes of one software module can be bundled into one data file by means of a Java tool JAR. The thus obtained data file is a so-called jar-file. Generally in the present method a class will contain a software routine or the like. Other embodiments could involve using classes containing reference to routines or the like and not the routine itself.

The receiver obtains the software module by combining the objects received from the sender with the class or group of classes either retrieved from its database or received from the sender. If the receiver succeeds in combining the objects and class or group of classes, the receiver transmits a message "transfer succeeded" to the sender. If the receiver cannot combine the objects and classes, a message "transfer not succeeded" will be transmitted to the sender.

In this manner data traffic in the network will be significantly reduced as generally it will only be necessary to transfer the objects of a software module.

Of course, at any transfer of a class or group of classes, a receiver receiving new classes can store the class or group of classes together with the corresponding unique class identifier in its database for future use. In this manner the database will gradually be further completed reducing future data traffic.

It is noted that although in the above-described exam-

ple, the objects and class identifier are transmitted from the sender to the receiver without interruption, its is also possible to transmit first the class identifier to check its presence at the receiver and thereafter the objects. However, the order described shows the advantage that the number of switchings from transmitting to receiving at the sender and receiver sides is reduced resulting in a further reduction of the time period for transferring a software module as switching in a computer network is relatively time consuming.

10 In the method of the invention the unique class identifier is preferably obtained in the following manner. First the user may give a predetermined name to a class or group of classes. This given name is a first part of the class identifier. Further, all classes of a software module are bundled
15 into one data file which is the data file to be transferred if the group of classes is absent at the receiver side. At the sender side a program is executed determining a cryptographic hash function of the data file of the group of classes and the result of this hash function is stored. This hash function re-
20 sult is the second part of the unique class identifier.

It is noted that any cryptographic hash function can be used which provides a result which is significantly shorter than the original data file, wherein a small change in the original data file provides a large change in the hash function result and wherein it is very difficult to determine an input data file leading to a predetermined hash function result.

As a further option to provide a unique class identifier, it is possible to add the length of the data file as a third part to the unique class identifier.

30 In this manner it is guaranteed that if a sender
transmits the unique class identifier of a software module to a
receiver and the receiver indicates that the class or group of
classes of the unique class identifier received is present in
its database, indeed the correct class or classes are available
35 at the receiver side.

Further, the use of a cryptographic hash function provides security to the transfer method. The receiver will always check the unique class identifier with the data file of classes

received. To this end the receiver will execute the same cryptographic hash function on the data file of the classes and will compare the result of its own hash function with the hash function result in the class identifier. If a match is not
5 found, the receiver will transmit a message "transfer not succeeded" to the sender and will terminate the transfer protocol.

The method described can be used advantageously in an Internet application of software agents, wherein the software agent travels on the Internet to search information and the
10 like. However, it will be understood that the method results in the same advantages of reducing data traffic between processes running in one computer system or the transfer of software modules between computers in any type of network.

The invention is not restricted to the above described
15 embodiment and can be varied in a number of ways within the scope of the following claims.

CLAIMS

1. Method for transferring a software module from a sender to a receiver in a computer system or network, wherein the software module comprises at least one object and at least one class, the object being an instance of the class(es), wherein each class or group of classes is provided with a class identifier, wherein both the sender and receiver comprises a database of classes and groups of classes with corresponding class identifier, wherein the sender transmits the class identifier of a software module to be transferred to the receiver and the receiver checks its database for presence of the received class identifier, and wherein the sender transfers only the object of the software module or both the object and class or group of classes depending on the presence or absence of the class or group of classes at the receiver, **characterised in that** the receiver transmits a message "present" or "absent" to the sender, and the sender provides each class or group of classes by combining a given name of each class or group of classes of a software module and the result of a cryptographic hash function, wherein said result is obtained by executing a cryptographic hash function on the data file of the class or group of classes.

2. Method according to claim 1, wherein the sender transmits first all objects and the class identifier to the receiver if a message "absent" is received.

3. Method according to claim 1 or 2, wherein the receiver obtains the software module to be transferred by combining the object received with the class or group of classes retrieved from its database or received, wherein the receiver transmits a message "transfer succeeded" or "transfer not succeeded" depending on whether or not the

8'

receiver succeeds in combining the object and class or group of classes.

4. Method according to any one of the preceding claims, wherein the receiver stores each class and group of
- 5 classes with the corresponding class identifier received in its database for later use.

graphic hash function on the data file of the class or group of classes.

56. Method according to ~~claim 5~~^{any one of the preceding claims}, wherein a sender further combines the length of the data file of the class or group of classes with the given name and the result of the hash function to provide the class identifier. ~~any one of the preceding claims~~

67. Method according to ~~claim 5 or 6~~⁶, wherein the receiver checks a class or group of classes received from a sender by comparing the result of the hash function of the received class identifier with the result obtained by carrying out the same cryptographic hash function on the data file of the class or group of classes received.

78. Method according to claim ~~7~~⁶, wherein the receiver transmits a message "transfer succeeded" or "transfer not succeeded" depending on the comparison of the result of the hash function on the data file received and the result of the hash function of the class identifier.

89. Method according to any one of the preceding claims, wherein senders and receivers are computers in computer network, such as the Internet.

910. Method according to claim ~~9~~⁸, wherein the software module is a so-called agent for searching, exchanging and/or providing information on the network.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau(43) International Publication Date
19 April 2001 (19.04.2001)

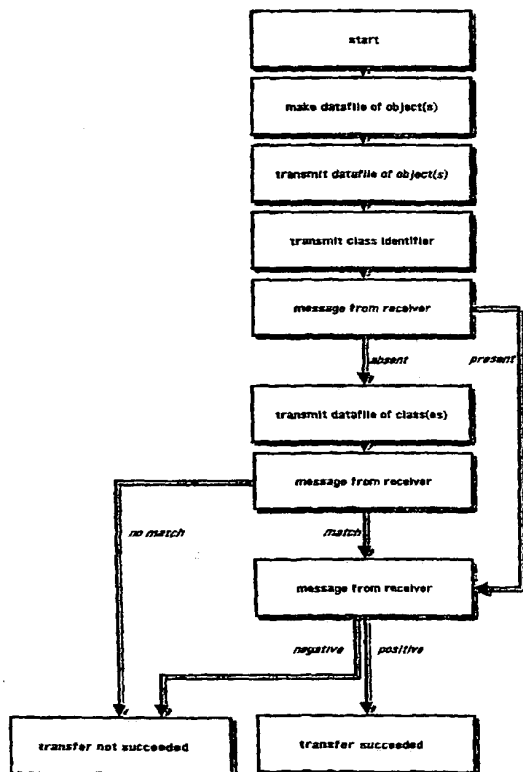
PCT

(10) International Publication Number
WO 01/27757 A1

- (51) International Patent Classification⁷: G06F 9/46 Joost [NL/NL]; Madridplantsoen 141, NL-2034 VS Haarlem (NL).
- (21) International Application Number: PCT/NL00/00720
- (22) International Filing Date: 6 October 2000 (06.10.2000)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
1013249 8 October 1999 (08.10.1999) NL
- (71) Applicant (for all designated States except US): TRYLIJAN BV [NL/NL]; Kruislaan 400, NL-1098 SM Amsterdam (NL).
- (72) Inventor; and
- (75) Inventor/Applicant (for US only): BOLWIDT, Erwin,
- (81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.
- (84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE,

[Continued on next page]

(54) Title: METHOD FOR TRANSFERRING A SOFTWARE MODULE FROM A SENDER TO A RECEIVER IN A COMPUTER SYSTEM OR NETWORK



(57) Abstract: In a method for transferring a software module from a sender to a receiver in a computer system or network, wherein the software module comprises at least one object and at least one class, the object being an instance of the class(es), each class or group of classes is provided with a class identifier. Both the sender and receiver comprises a database of classes and groups of classes with corresponding class identifiers. The sender transmits the class identifier of a software module to be transferred to the receiver and the receiver checks its database for presence of the received class identifier. The receiver transmits a message "present" or "absent" to the sender and the sender transfers only the object of the software module or both the object and the class or group of classes depending on the presence or absence of the class or group of classes at the receiver.

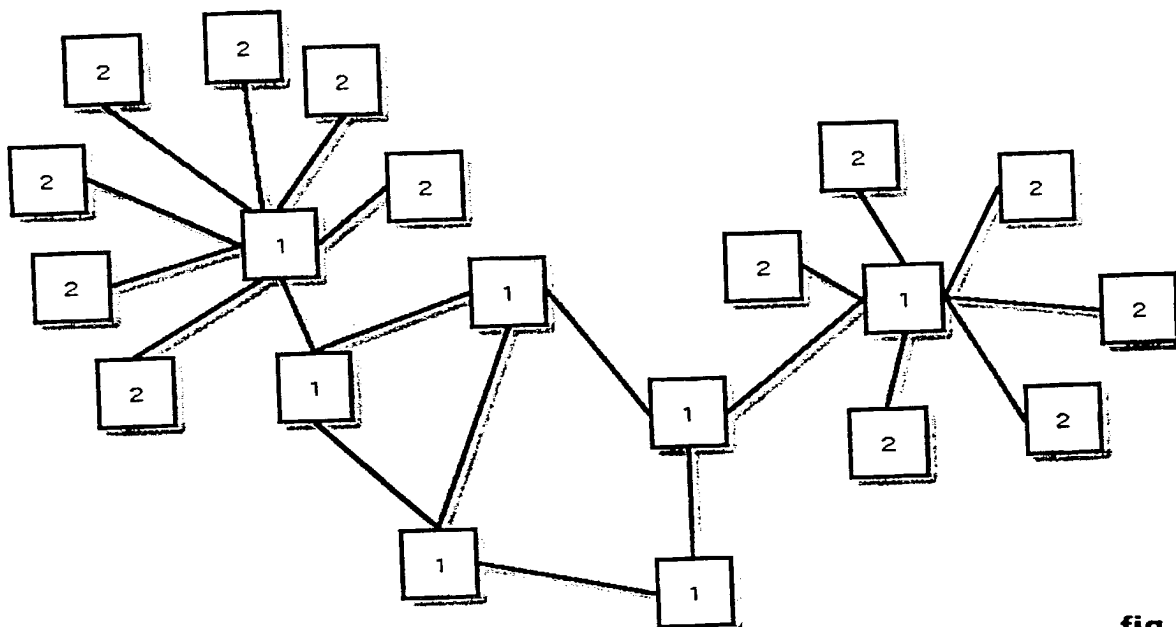


fig.1

2/3

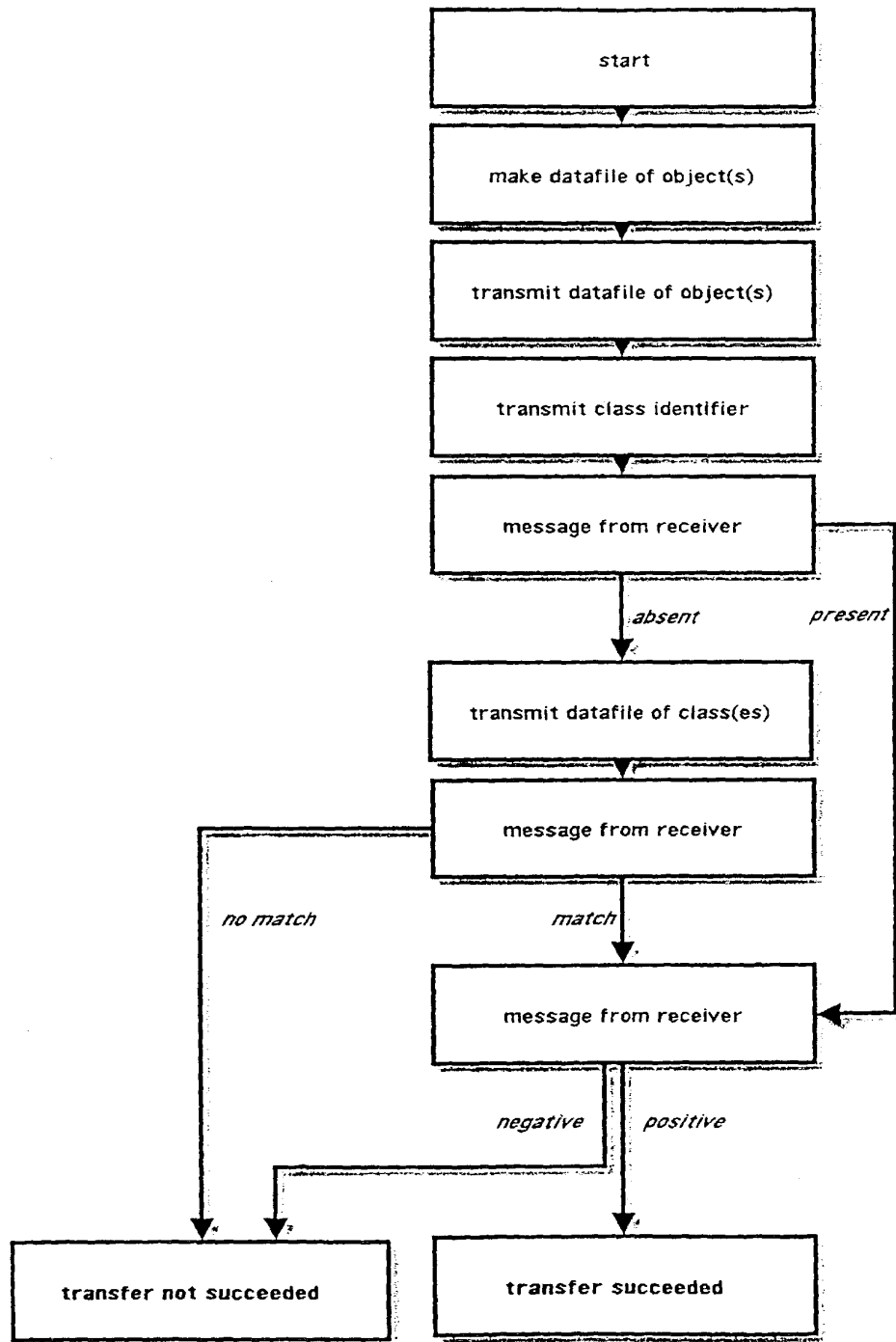


fig.2

3/3

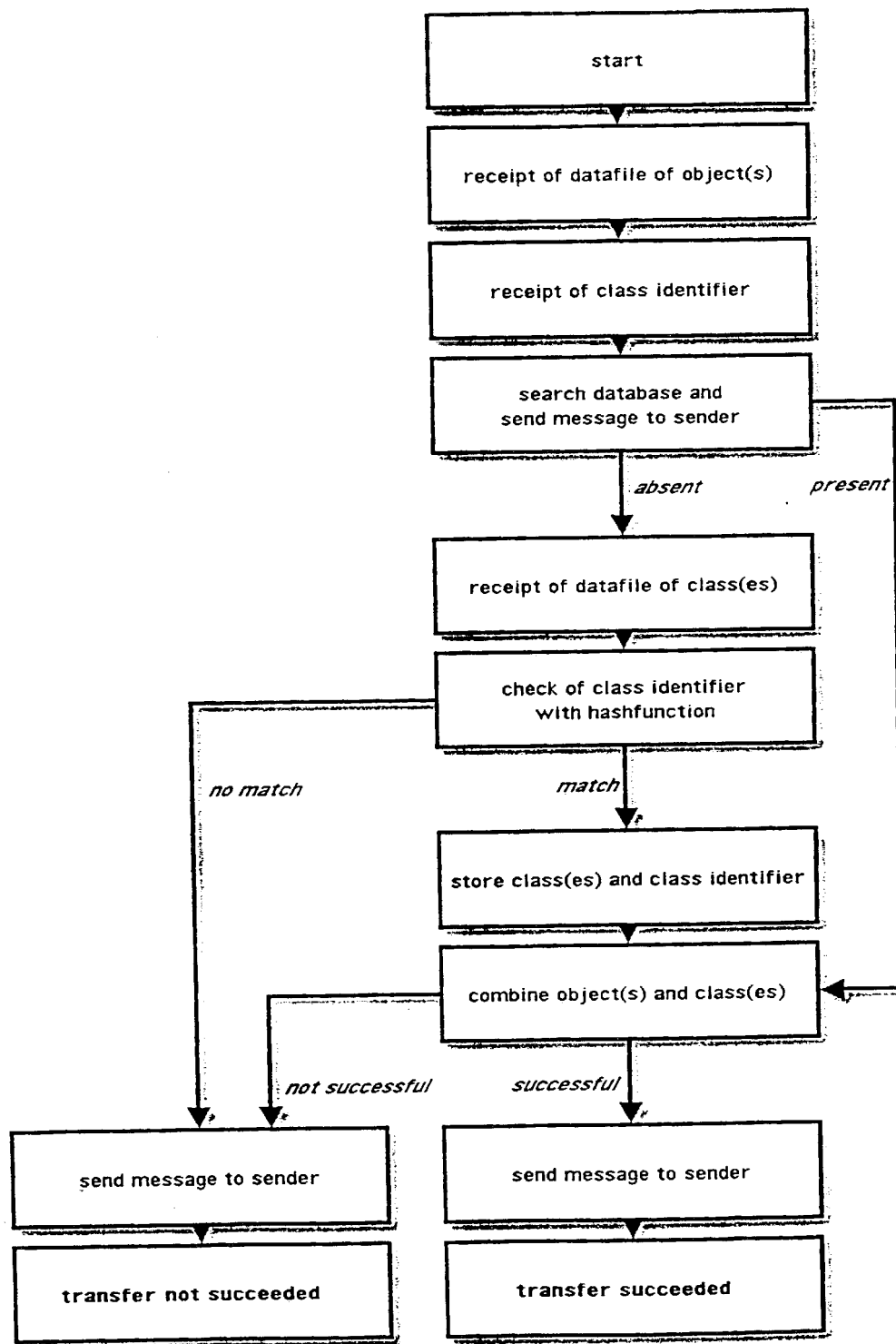


fig.3

Attorney Docket No. DVME-1024US

DECLARATION AND POWER OF ATTORNEY

As below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name,

I believe I am the original, first and sole inventor (if only one name is listed below) or an original first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

**Method for Transferring A Software Module From A Sender to a Receiver In A
Computer System or Network**

The specification of which

☐ is attached hereto.

☒ was filed on 06 October 2000 as United States Application No. or PCT
International Application Number PCT/NL00/00720 and was amended on 25 October 2001.

(if applicable)

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119(a)-(d) or Section 365(b) of any foreign application(s) for patent or inventor's certificate, or Section 365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate or PCT International having a filing date before that of the application on which priority is claimed.

Prior Foreign Application(s)			Priority Not Claimed
1013249 /	Netherlands /	08 October 1999 /	<input type="radio"/>
(Number)	(Country)	(Day/Month/Year Filed)	
			<input type="radio"/>
(Number)	(Country)	(Day/Month/Year Filed)	
			<input type="radio"/>
(Number)	(Country)	(Day/Month/Year Filed)	

I hereby claim the benefit under 35 U.S.C. Section 119(e) of any United States provisional application(s) listed below:

(Application Serial No.)	(Filing Date)
(Application Serial No.)	(Filing Date)
(Application Serial No.)	(Filing Date)

I hereby claim the benefit under 35 U.S.C. Section 120 of any United States application(s) or Section 365(c) PCT application designating the United States, listed below; and insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International Application in the manner provided by the first paragraph of 35 U.S.C. Section 112, I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in Title 37 C.F.R. Section 1.56 which became available between the filing date of the prior application and the national or PCT International filing date:

(Application Serial No.)	(Filing Date)	(Status) (patented, pending, abandoned)
(Application Serial No.)	(Filing Date)	(Status) (patented, pending, abandoned)
(Application Serial No.)	(Filing Date)	(Status) (patented, pending, abandoned)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements any jeopardize the validity of the application or any patent issued thereon.

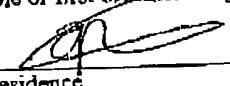
11/11 '02 11:30 FAX 0031205110931
11/11/2002 11:01 020-8884326

DE VRIES & METMAN
IRYLLIAN BV

KNOBLE

0006

4 -
POWER OF ATTORNEY: As named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.
John J. Knoble, Registration No. 32,387; Ken I. Yoshida, Registration No. 37,009; Kevin J. Dunleavy, Registration No. 32,024 and Jianzhong Shen, Registration No. 48,076 of the firm KNOBLE & YOSHIDA, LLC, Eight Penn Center, 1628 John F. Kennedy Blvd., Philadelphia, PA 19103, Telephone: (215) 599-0600, Facsimile: (215) 599-0601. Please direct all correspondence and telephone calls to Kevin J. Dunleavy,

Full name of sole or first inventor	
1-00 <u>Erwin Joost Bolwidt</u>	Date
Sole or first inventor's signature 	<u>9 November 2002</u>
Residence <u>Haarlem, The Netherlands NLX</u>	
Citizenship <u>The Netherlands</u> ✓	
Post Office Address <u>Madridplantsoen 141, NL-2034 VS Haarlem, The Netherlands</u>	